IN THE CLAIMS:

1. (Currently Amended) A [keystore] method of managing a composite keystore generated from user local keystores comprising the steps of:

retrieving one or more [certificate]certificates from a local database of a user, wherein the certificates are associated with the user;

responsive to retrieving said one or more certificates, determining if [said] any of said one or more certificates preexists in a preselected portion of a distributed database of the composite keystore; and

storing nonpreexisting certificates of said one or more certificates in said preselected portion of said distributed database.

2. (Original) The method of claim 1 wherein said preselected portion of said distributed database comprises said distributed database.

3. (Currently Amended) The method of claim 1 further comprising, responsive to determining that said one or more certificates do not preexist in the preselected portion of the distributed database, [the step of] determining if said one or more certificates is invalid.

4. (Currently Amended) The method of claim 3 wherein said step of storing nonpreexisting [[ones]]certificates of said one or more certificates is bypassed for invalid certificates.

5. (Currently Amended) The method of claim 3 further comprising the step of requesting, by the composite keystore, a new certificate corresponding to an invalid certificate, wherein the new certificate is valid and corresponds to the invalid certificate.

6. (Original) The method of claim 1 further comprising the step of updating said distributed database in response to an update event.

7. (Currently Amended) The method of claim 6 wherein said step of updating said distributed database comprises the steps of:

requesting one or more new certificates by the composite keystore; and

adding said new certificates to said distributed database.

8. (Currently Amended) The method of claim 1 further comprising the steps of:

responsive to determining any of said one or more certificates preexists, determining if a current certificate supercedes a preexisting certificate; and

responsive to determining the current certificate supercedes the preexisting certificate, replacing said preexisting certificate with said current certificate if said current certificate supercedes said preexisting certificate.

9. (Currently Amended) The method of claim 1 further comprising the steps of:

accessing said distributed [keystore]database; and

responsive to accessing said distributed database, requesting a selected certificate from said distributed [keystore]database, wherein the selected certificate is used for encrypting data to be transferred in a secured data transfer.

10. (Currently Amended) The method of claim 9 further comprising the step of searching [[a]]the local [keystore]database for said selected certificate in response to a failure of said step of requesting said selected certificate.

11. (Currently Amended) The method of claim 1 further comprising the step of repeating, for a second local database, the steps of:

retrieving one or more certificates from the second local database of a second user;

determining if [[said]] any of said one or more certificates retrieved from the second local database preexists in a second preselected portion of [[a]]the distributed database; and

storing nonpreexisting certificates of said one or more certificates from the second local database in said second preselected portion of said distributed database.

12. (Original) The method of claim 8 wherein said distributed database comprises a logical keystore.

13. (Currently Amended) A computer program product embodied in a tangible storage medium, the program product for managing a composite keystore generated from user local keystores, the program product including a program of instructions for performing the steps of:

retrieving one or more certificates from a first local database of a user, wherein the certificates are associated with the user;

responsive to retrieving said one or more certificates, determining if [[said]] any of said one [[o]]or more certificates preexists in a preselected portion of a distributed database of the composite keystore; and

storing nonpreexisting certificates of said one or more certificates in said preselected portion of said distributed database.

14. (Original) The program product of claim 13 wherein said preselected portion of said distributed database comprises said distributed database.

15. (Currently Amended) The program product of claim 13 wherein said program of instructions further comprises programming for performing, responsive to determining that said one or more certificates do not preexist in the preselected portion of the distributed database, the step of determining if said one or more certificates is invalid.

16. (Currently Amended) The program product of claim 15 wherein said step of storing nonpreexisting [[ones]]certificates of said one or more certificates is bypassed for invalid certificates.

17. (Currently Amended) The program product of claim 15 wherein said program of instructions further comprises programming for performing the step of requesting, by the

composite keystore, a new certificate corresponding to an invalid certificate, wherein the new certificates is valid and corresponds to the invalid certificate.

18. (Original) The program product of claim 13 wherein said program of instructions further comprises programming for performing the step of updating said distributed database in response to an update event.

19. (Currently Amended) The program product of claim 18 wherein said step of updating said distributed database comprises the steps of:

requesting one or more new certificates by the composite keystore; and

adding said new certificates to said distributed database.

20. (Currently Amended) The program product of claim 13 wherein said program of instructions further comprises programming for performing the steps of:

responsive to determining any of said one or more certificates preexists, determining if a current certificate supercedes a preexisting certificate; and

responsive to determining the current certificate supercedes the preexisting certificate, replacing said preexisting certificate with said current certificate if said current certificate supercedes said preexisting certificate.

21. (Currently Amended) The program product of claim 13 wherein said program of instructions further comprises programming for performing the steps of:

accessing said distributed database; and

responsive to accessing said distributed database, requesting a selected certificate from said distributed database, wherein the selected certificate is used for encrypting data to be transferred in a secured data transfer.

22. (Currently Amended) The program product of claim 21 wherein said program of instructions further comprises programming for performing the step of searching [[a]]the local [keystore]database for said selected certificate in response to a failure of said step of requesting said selected certificate.

23. (Currently Amended) The computer program product of claim 13 wherein said program of instructions further comprises instructions for the step of repeating, for a second local database, the steps of:

retrieving one or more certificates from the second local database;

determining if [[said]] any of said one or more certificates from the second local database preexists in a second preselected portion of [[a]]the distributed database; and

storing nonpreexisting certificates of said one or more certificates from the second local database in said second preselected portion of said distributed database.

24. (Original) The computer program product of claim 20 wherein said distributed database comprises a logical keystore.

25. (Currently Amended) A data processing system for managing a composite keystore generated from user local keystores comprising:

circuitry operable for retrieving one or more certificates from a first local database of a user, wherein the certificates are associated with the user;

circuitry operable for determining, responsive to retrieving said one or more certificates, if [[said]] any of said one or more certificates preexists in a preselected portion of a distributed database of the composite keystore; and

circuitry operable for storing nonpreexisting certificates of said one or more certificates in said preselected portion of said distributed database.

26. (Original) The system of claim 25 wherein said preselected portion of said distributed database comprises said distributed database.

27. (Currently Amended) The system of claim 25 further comprising circuitry operable for, responsive to determining that said one or more certificates do not preexist in the preselected portion of the distributed database, determining if said one or more certificates is invalid.

28. (Currently Amended) The system of claim 27 wherein said circuitry operable for determining if said one or more certificates is [expired]invalid includes circuitry operable for bypassing, for invalid certificates, said circuitry operable for storing nonpreexisting certificates.

29. (Currently Amended) The system of claim 27 further comprising circuitry operable for requesting, by the composite keystore, a new certificate corresponding to an invalid certificate.

30. (Original) The system of claim 25 further comprising circuitry operable for updating said distributed database in response to an update event.

31. (Currently Amended) The system of claim 30 wherein said circuitry operable for updating said distributed database comprises:

circuitry operable for requesting one or ore new certificates by the composite keystore; and

circuitry operable for adding said new certificates to said distributed database.

32. (Currently Amended) The system of claim 25 further comprising:

circuitry operable for, responsive to determining any of said one or more certificates preexists, determining if a current certificate supercedes a preexisting certificate; and

circuitry operable for replacing, responsive to determining the current certificate supercedes the preexisting certificate, said preexisting certificate with said current certificate if said current certificate supercedes said preexisting certificate.

33. (Currently Amended) The system of claim 25 further comprising:

circuitry operable for accessing said distributed database; and

circuitry operable for requesting, responsive to accessing said distributed database, a selected certificate from said distributed database, wherein the selected certificate is used for encrypting data to be transferred in a secured data transfer.

34. (Currently Amended) The system of claim 33 further comprising circuitry operable for searching [[a]]the local [keystore]database for said selected certificate in response to a failure of said step of requesting said selected certificate.

35. (Currently Amended) The system of claim 25 further comprising circuitry operable for repeating, for a second local database of a second user, the steps of:

retrieving one ore more certificates from the second local database;

determining if [[said]] any of said one or more certificates of the second local database preexists in a second preselected portion of [[a]]the distributed database; and

storing nonpreexisting certificates of said one or more certificates from the second local database in said second preselected portion of said distributed database.

36. (Original) The system of claim 32 wherein said distributed database comprises a logical keystore.